Exhibit C

Flowcharting With the ANSI Standard: A Tutorial

NED CHAPIN

InfoSci Inc., Menlo Park, California

The ISO and ANSI X3.5 standard flowchart symbols and their usage in information processing are explained and examples given. The two main categories of flowchart—the system chart or run diagram, and the flow diagram or block diagram—are stressed. For each, the outline symbols and their manner of use are presented, as well as guidelines and conventions, such as cross-referencing. In the case of flow diagrams, notation is presented for use within the outline symbols.

Key words and phrases: standards, flowchart, flow diagram, system chart, run diagram, block diagram, program flowchart, documentation, outlines, boxes, notation, symbols, program description, system description, algorithm statement, communication

CR categories: 1 3, 2.2, 2.43, 4.0

HISTORICAL DEVELOPMENT

Flowcharting is a means of graphically stating ways of solving information handling problems. Flowcharting, as people use the term in working with computers, must be distinguished from other graphic aids. For stating clerical procedures, such as those used in systems and procedures work, people use a graphic means which has also been the subject of a standard [4]. But it is quite different from the standard under discussion here. Logic designers also use graphic aids for stating the character of the machines they design for handling information. These too have been the subject of a standard [3, 19]. The emphasis in this tutorial paper is on stating information handling problems where the information handling is done at least in major part with the aid of the automatic computer [7].

The intellectual father of flowcharting is John von Neumann. He and his associates at Princeton University's Institute for Advanced Study were the first to use graphic aids systematically for this purpose and publish their use [11]. Even though the details of the flowcharting as the standard specifies it today differ considerably from what they advocated, the spirit, the philosophy, and the rationale remain much as they presented them.

For their own internal purposes and for dealing with customers, each of the major computer manufacturers has over the course of the years developed, adopted, published, modified, and advocated flowcharting conventions (see, for example, [13, 15]). These have differed from vendor to vendor, in part deliberately as an attempt to distinguish one vendor from the competing vendors, and in part out of a sincere attempt to reflect what each has felt to be unique differences in their philosophy and approach to information processing problems.

Users of computers have individually and collectively made decisions on flowcharting conventions. Most small and medium and many large computer users have adopted the conventions presented to them by the vendor of the computer they

Copyright © 1970, by Ned Chapin

CONTENTS

have elected. But a few larger users have chosen to go their own way, and have developed their own internal standards. The United States Air Force, for example, has developed its own standards for this purpose and in practice has had a multiplicity of them (see, for example, [21]).

Users of computers acting collectively through the user groups have sometimes addressed themselves to the problem of standards for flowcharting. Since these user groups normally have been composed of users of only one vendor's computer, the effect usually has been to recommend modifications or suggestions to the computer vendor for changes in the vendor's standard. But a few users' groups, for example SHARE, have independently presented their own standards and have advocated them for general adoption [10].

Individuals with competence and standing in the computer field have for their own use sometimes deviated from the practice of the various computer vendors and from the other sources. They have presented their own recommendations for flowcharting. These recommendations have been effective through their authors' publications (see, for example, [7, 12, 16, 18]). Some of these recommendations, however, have ceased to be significant with the advent of formal standardization.

Using the computer itself to produce flowcharts has had relatively little influence on the development of standards, although programs for doing it have been in use since 1957.[1] To assist the process, a Systems Flowchart Language (SFL) was even developed [17]. Of more practical significance in popularizing the *de facto* vendor standards has been the vendors' practice of providing free plastic templates to aid in drawing flowcharts (see, for example, [14]).

During the 1960s a committee attempted to develop a standard for flowcharting. Working through the Business Equipment Manufacturers Association and the Ameri-

[1] The author has pending publication a paper on the computer production of flowcharts. It includes a summary of the historical development of these techniques.

can Standards Association,[2] with the committee members drawn from computer vendors and a few major computer users, the committee after the usual compromises drew up a proposed standard and circulated it for reaction. With revisions, it was approved in 1963 and published as an American Standard. The Association for Computing Machinery and other groups published this standard in their periodicals, giving it considerable publicity (see, for example, [1]). This standardization effort in the United States paralleled a similar effort conducted for the International Standards Organization (ISO).

Subsequently, in 1965 and again in 1966, 1968, and 1970, the American Standard was revised. The 1965 revision was major, but the 1966 and 1968 revisions were only minor. The 1970 revision extended the standard to match more closely the ISO standard. The standard as it is presented in this tutorial paper is the ANSI 1970 revision [2]. Since this is a tutorial paper, it does not cover every part of the standard, nor every variation in the use of the standard.

ANSI STANDARD

Outlines

The ANSI standard consists, in the first place, of a series of graphic outlines or boxes, which the standard terms "symbols." The standard advances these flowchart outlines in three groups: the basic, the additional, and the specialized. Complete flowcharts can be drawn using only the basic and the additional outlines. The use of the specialized outlines is optional. If they are used, however, they should be used in a manner consistent with the standard.

For the outlines in each group, the standard specifies the shape, but not the size. The shape is specified in two ways: by the ratio of the width to the height and by the general geometric configuration.

[2] As of 1969, the American National Standards Institute (ANSI), 1430 Broadway, New York, N.Y. 10018.

This means that the user of the standard is free to draw outlines of any size to fit his own convenience. He may vary the size from time to time during the course of his flowchart, but he is to observe the ratio and general configuration specified. For those who do not have ready access to the standard, the Appendix to this paper summarizes the ratios and configurations.

The standard implicity advances the use of a single width or weight of line for drawing the outlines. It also implicitly advances a single orientation or positioning of the outlines with respect to each other. In particular, portions of the outlines shown horizontally oriented in the standard are to be drawn that way.

Basic Outlines

The basic outlines specified in the standard are the input-output, the process, the flowline, and the annotation outlines. These are illustrated in Figure 1 and described in the Appendix.

The input-output outline indicates an input or output operation, or input or output data. It is defined for use irrespective of media, format, equipment, and timing. Some specialized outlines may be substituted for this outline.

The process outline is the general purpose outline. It is the *de facto* default outline for use when no other outline is specified by the standard. The process outline indicates data transformation, data movement, and logic operations. Some specialized outlines may be substituted for this outline.

The flowline outline is an arrow of any length which connects successive other outlines to indicate the sequence of operations or data (the "direction of flow"). It is defined for use in an alternating fashion with the other outlines. As such, it also indicates the sequence in which the other outlines are to be read. To specify the direction of flow or reading, open arrowheads may be used on any flowline as shown in Figure 1.

The normal direction of flow is the normal direction of reading for people

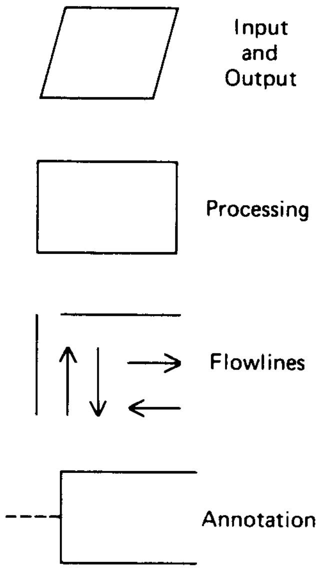

FIG. 1. Basic outlines

trained in the English language: from top to bottom and from left to right. Where the flow follows this normal pattern, no open arrowheads are needed to remind the reader. In the event of any significant deviation from this pattern, arrowheads are required to signal the deviation to the reader's attention. Whenever the direction of flow might be ambiguous to a reader, arrowheads should be used to provide clarification. Bidirectional flow may be indicated by dual arrows each with open arrowheads, or less preferably by open arrowheads in both directions on single flowlines.

The annotation outline provides a way to supply descriptive information, comments, and explanatory notes. Its dashed line indicates the outline to which this explanation or clarification applies.

Additional Outlines

The additional outlines are for the convenience of the reader, and not for the purpose of describing data-processing action. These symbols provide for handling the limitations of pages of various sizes, and make it more convenient to show connections in the sequences of flow. These outlines are shown in Figure 2.

The connector outline, a circle, must in practice be used at least in pairs. To that end, the standard advances two varieties, the inconnector or entry connector, and the outconnector or exit connector. An inconnector or entrance has a flowline leaving it but none entering it; an outconnector or exit has a flowline entering it but none leaving it. Each inconnector may have from zero through any number of outconnectors associated with it. However, each outconnector must have exactly one inconnector associated with it. One function of the connector outline is to enable a long sequence of outlines (a "flow") to be broken into pieces to fit conveniently on a page. The connector outline also provides ways of joining together convergent lines of flow that fan in to some particular point. And, it provides a way of identifying divergent lines of flow.

The terminal connector outline serves to indicate a beginning, an end, or a break in the usual line of flow. In the first two uses, it substitutes for an ordinary connector at the beginning and the end of major portions of a sequence of outlines (a "flow"), particularly when these portions are identified by a name, as, for example, for a closed subroutine. In its

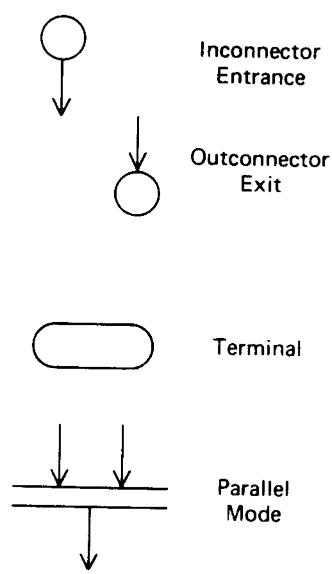

FIG. 2. Additional outlines

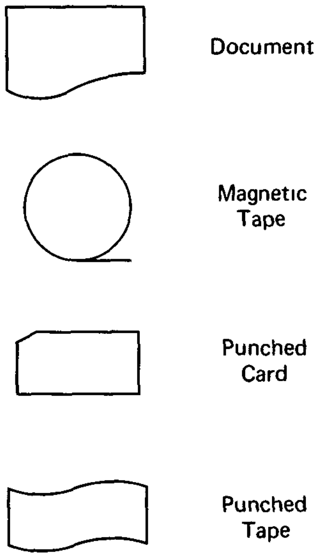

FIG. 3. Specialized outlines for media

third use, it may represent a start, stop, halt, delay, pause, interrupt, or the like. For this use, it has both an entry and an exit flowline.

The parallel mode outline is a pair of horizontal lines with one or more vertical entry flowlines and one or more vertical exit flowlines. It is used to indicate the start or end of simultaneous operations.

Specialized Outlines

Groups. The specialized outlines fall into three groups. One group permits specification of the data-carrying media (see Figure 3). Another permits specification of the peripheral equipment type (see Figure 4). A third permits specification of selected types of processing action (see Figure 5). In each case, where no specialized outline has been provided, the standard specifies that the basic outline covering the situation should be used. Thus, for any media or equipment, it should be the input-output outline. For any processing, it should be the process outline. The one exception is the communication link, for which the basic outline is the flowline.

Media outlines. The document outline is the most commonly used of all the specialized media outlines. This outline, a stylization of a torn piece of paper, represents data in the form of hard copy input or output of any type. For example, it may represent data taking the form of printing on paper produced by a high speed printer, or of marks on cards read by an optical reader, or of a graph produced by a data plotter, or of a page of typing produced on a terminal.

The magnetic tape outline is a circle

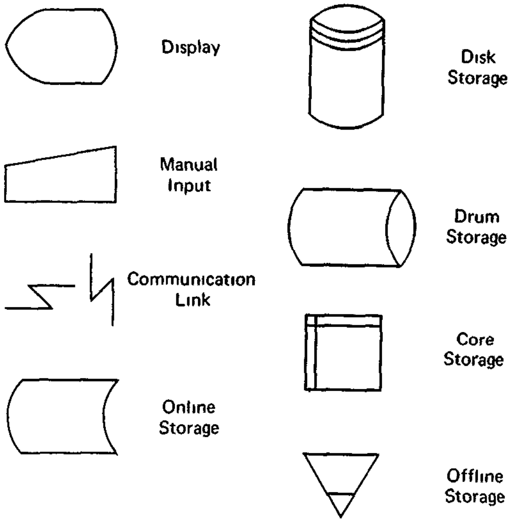

FIG. 4. Specialized outlines for equipment

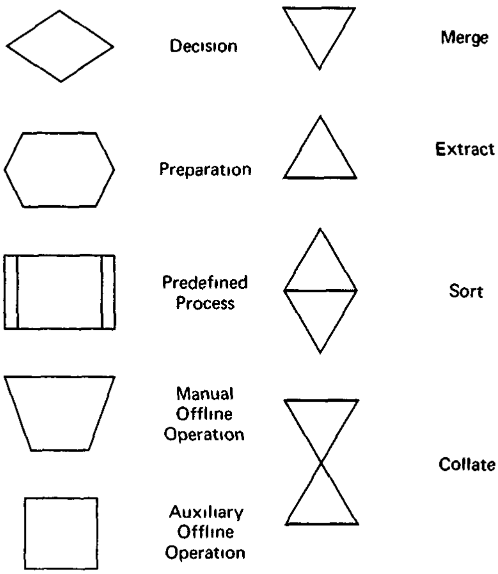

FIG. 5. Specialized outlines for processing

with a horizontal rightward pointing line tangent to the bottom. This outline represents data in the medium of magnetic tape.

The punched-card outline represents data in the medium of a punched card of any style, size, or punching, such as Hollerith punched cards, binary punched cards, Binary Coded Decimal (BCD) punched cards, fifty-one column cards, and the like. Thus, a time card which only has on it printed numbers and never is punched with equivalent or even unrelated information does not qualify for representation with the punched-card outline (it takes a document outline instead). Two further specialized forms of the punched-card outline are described below.

The punched-tape outline represents data in the medium of any punched continuous material, such as paper tape, punched plastic tape, punched metal tape, etc. The requirements to be met are that the medium be indefinite in length and that the data be represented by punch patterns. The outline is a stylization of a partial fold of paper tape.

Equipment outlines. The display output outline is a stylization of a cathode-ray tube (CRT) with the face of the tube to the right and the neck of the tube to the left. This outline represents any kind of transitory data not in hard copy form, as for example CRT displays, console displays, and the like. But the standard also advances the display outline for intermediate output data used during the course of processing to control the processing. Common examples are the data produced on console printers and time-sharing terminals if the human user is expected to utilize immediately the data presented.

The manual input outline represents data acquired by human control of manually operated online equipment. Examples are data from the operation of keyboards, light pens, console switch settings, push-buttons, transaction recorders, tag readers, and the like, where the human operator provides the timing.

The communication link outline is represented by a zigzag flowline. This is appropriate because typically data communication done with the aid of equipment provides a flow of data from one place, or from one medium or equipment, to another. As such, even though the outline is equipment oriented, it is used as a specialized flowline. Where necessary, open arrowheads may be used to indicate the direction of flow, in the same manner described previously.

The on-line storage outline represents data held in any on-line intermediate and external storage device of any type, as for example magnetic disks, magnetic tapes, magnetic drums, magnetic cards, additional banks of magnetic core storage, microfilm, etc. For data on some of these devices, the standard provides more specialized outlines when more precise specification is desired, as described below. A more specialized outline for data on magnetic tape was noted above in the media group.

The disk storage outline represents data stored on a disk device of any type, especially a magnetic disk. The outline is a stylization of a cylinder standing on end. The outline is a further specialization of the on-line storage outline.

The drum storage outline represents data stored on a drum device, especially a magnetic drum. The outline is a stylization of a drum lying on its side. The outline is a further specialization of the on-line storage outline.

The core storage outline represents data stored in a magnetic core or similar high speed device that is *not* the primary internal storage for the computer. It might be, for example, an auxiliary on-line bulk core device, or a remote computer connected on-line with the computer doing the main processing. The outline is a stylization of two drive lines in a magnetic core array.

The off-line storage outline is an equilateral triangle with a small bar. This outline represents any data stored off-line regardless of the medium and regardless

of the equipment used. In common practice, it is used for manually maintained data files.

Process outlines. The most common of the specialized process outlines is the decision outline. It indicates comparison, deof alternative flows (sequences of operawhich determine or select among a variety of alternative flows (sequences or operations). As such, the number of flowlines leaving a decision outline must always be greater than one.

Sometimes the number of flowlines leaving a decision outline (the number of exits) exceeds three. In this case, the standard advances, as shown in Figure 6, several alternatives which are equally acceptable. One is the organizational chart tree pattern of flowlines from a single flowline leaving a decision outline. Another alternative used in the same way is a vertical flowline which has a number of horizontal flowlines from it. To save space, formal outconnectors may be omitted from these two forms. Quite a different alternative is a "branching" table in the form of a series of pairs of rectangles, packed together in a double row or a double column. The upper or left portion replaces the decision outline; the lower or right portion replaces the usual outconnectors. The tables use their first pairs of boxes for explanation of the branching.

The preparation outline indicates operations on the program itself. They are usually control, initialization, cleanup, or overhead operations not concerned directly with producing the output data, but usually necessary to have done. Three examples are setting the limiting value as an iteration control, decrementing an index register, and setting the value of a program switch. By convention, when either a decision or a preparation outline could be used (such as for testing a switch), the common practice is to use the decision outline.

The predefined process outline indicates or identifies one or more operations which are specified in more detail elsewhere, as in a booklet or in a different flowchart (but not in another part of this same flowchart). Examples of a predefined process are a named closed subroutine or a routine from the operating system for the computer.

The manual operation outline indicates any off-line input or output producing operation which has its speed determined by the speed of the human operator, as for example entering data off-line by means of a keyboard as in a keyboard-to-magnetic-tape operation, or finding a folder in a file cabinet drawer.

The auxiliary operation outline indicates any off-line operation performed on equipment which operates at its own speed or a speed determined by something other than the speed of its human operator. Examples of auxiliary operations are card-sorting operations, punched-card interpreting operations, and the like. Auxiliary operations are performed by equipment, not by human beings.

The merge outline indicates the creation of one set of items from two or more sets having the same sort sequences. The outline may be used for both on-line and off-line operations.

The extract outline indicates the reverse of the merge. That is, it indicates the creation of two or more sets of items from and in the same sort sequence as the original set. The outline may be used for both on-line and off-line operations.

The sort outline indicates the sorting of a set of items into some sequence on the basis of some (usually specified) key. The outline may be used for both on-line and off-line operations.

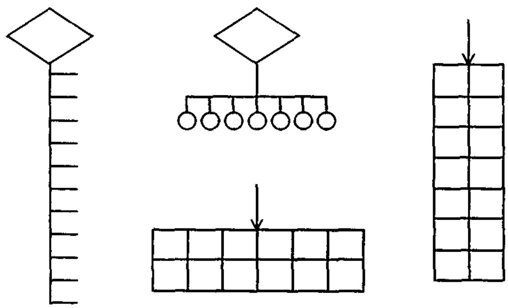

Fig 6. Outlines for large numbers of decisions

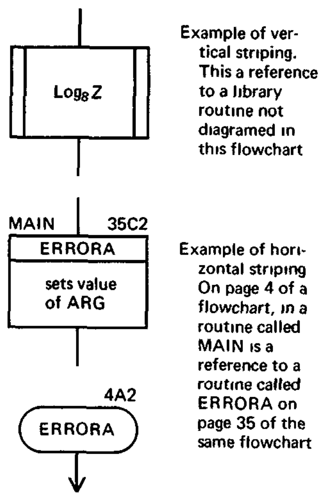

FIG. 7. Conventions for striping and references

The collate outline indicates a combination of merge and extract. Thus, this outline requires more than one entrance flowline and more than one exit flowline. This definition of collate used in the standard is not fully consistent with the usual definition of collate. The outline may be used for both on-line and off-line operations.

Standard Conventions

Striping. The standard specifies limited use for either horizontal or vertical striping within an outline. The vertical striping has already been covered in the special outline for predefined process (see Figure 7). Other uses of vertical striping are not specified by the standard.

Horizontal striping is advanced by the standard as one alternative way of indicating a reference to another part of the flowchart which provides a more detailed representation, as for example, of a subroutine. A horizontal line may be drawn from the left edge to right edge in the upper portion of an outline, except for the flowline, communication link, and additional outlines. The upper area thus enclosed is used to refer to some other part of the flowchart. The lower enclosed area is used in the usual manner, as shown in Figure 7. Wherever a horizontal striping is used within a symbol, the portion of the flowchart referred to must in turn be represented on the flowchart as beginning with and ending with terminal outlines as described previously. Both the detailed representation and the striped outline must have location cross-references, as described below.

Cross-references. In order to make cross-referencing easy between parts of the flowchart, the standard advances two conventions. One is to use or to assign names to portions of the flow represented by the flowchart. These names often are the same names used in the program or system. These may be the same as, or different from, the identifying names used for connectors.

An alternative convention (not mutually exclusive with the other) is to identify a location on each physical piece of the flowchart, as for example, in terms of page, row, and column, as in the manner of map coordinates. An example is the reference to page 4, row A, column 2 cited for the terminal outline in Figure 7. The standard leaves open the exact manner of composing such location references.

The ANSI standard is still in conflict with the ISO standard and with previous usage in the United States on the handling of references. The ISO and general American usage has been to place the identifying name immediately above and to the left of the outline (such as MAIN in Figure 7), and to place the coordinate reference above and immediately to the right of an outline (such as 35C2 in Figure 7). The ANSI standard advances exactly the opposite convention, but recognizes and cites the deviation from the ISO standard. In this paper, the ISO convention is used since it is also a common usage in the United States and since the ANSI standard explicitly recognizes the ISO position.

Crossing flowlines. The standard makes specific provision for connectors and cross-references. These can be used to avoid the necessity of having crossing flowlines.

If it is desired to use crossing flowlines, then the standard specifies that the flow-

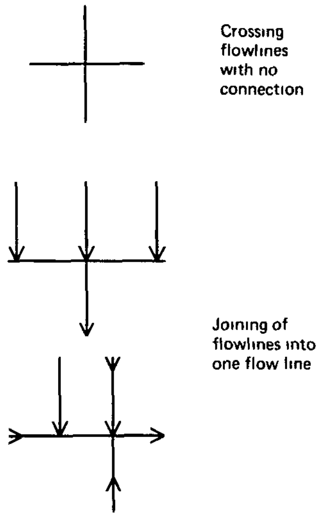

FIG. 8. Conventions for flowlines

lines shall have no arrowheads in the vicinity of the crossover. The standard specifies that the presence of arrowheads on the flowlines is to indicate a conjunction or coming together of the flow. The direction of flow at such points of joining flowlines is to be designated by the position and direction of the arrowheads, as shown in Figure 8.

Multiple outlines. The standard makes specific provision for the possibility of multiple instances for the specialized media outlines. These take two forms, one specifically for punched-card media and the other for media generally. The one applicable to punched cards only advances a convention for representing a deck of cards and a card file, as shown in Figure 9. These find their most common use in flowcharts for systems implemented with punched-card handling equipment.

The standard also advances a more general convention for use when multiple forms of specified media have different identifications and uses. For example, in punched-card installations, it is common to have a master or header card followed by several detail or trailer cards. To represent this situation, the standard advances that the main or first medium outline should be drawn in full. Following it and partially obscured by it in any clockwise position from it may be drawn in sequence a partial but closed outline of other instances of the same medium, as shown in Figure 9.

The standard is not clear with regard to the handling of flowlines for multiple outlines. The standard advances the convention that flowlines may enter or leave from any part of the multiple outline group. This in effect treats the group as though it were a single outline simply having inner lines marking off interior portions. But for those common cases where the group is to be broken into component parts and the component parts processed differently, the standard is silent. Perhaps the designers of the standard assumed all group formation and separation to be operations requiring a process outline, although this is contrary to long-standing custom.

To avoid this problem, this tutorial paper is based on the position that if a single flowline enters and a single flowline exits, then the multiple media shown are treated as a group. If multiple flowlines are used on either the entrance or exit sides, then the multiple flowlines apply only to the specific media outlines to which they individually connect.

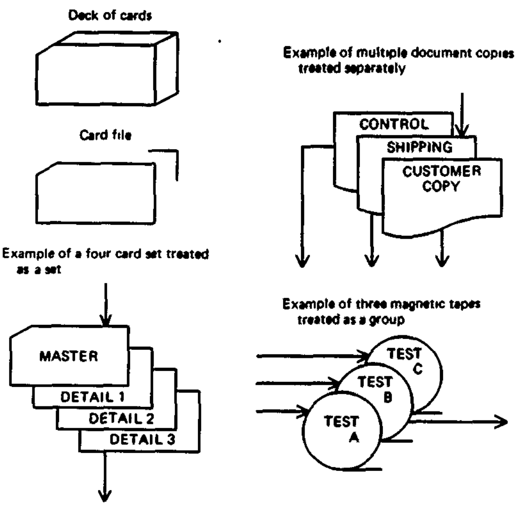

FIG 9 Multiple symbol conventions

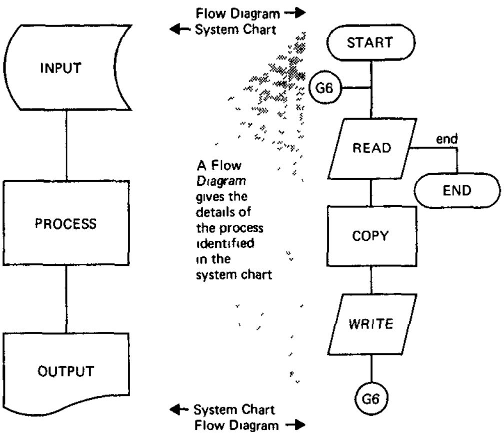

FIG 10. Relationship between flow diagram and system chart

USE OF THE STANDARD

Situations

The ANSI standard flowchart symbols for information processing cover two major situations. One situation is for representing algorithms, especially those for execution by a computer. The other is for representing systems without indicating the character of the component algorithms. Some other situations are noted briefly later in this paper. The term "flowchart," as used in the standard, may therefore refer to either of these situations.[3] Hereafter in this paper a clear distinction is necessary between flowcharts of systems and flowcharts of algorithms. Hereafter, "flow diagram" designates a flowchart of an algorithm, and "system chart" designates a flowchart of a system.

Other terms are also current in the field for these two situations. Thus other terms sometimes used for flow diagram are block diagram, logic chart, and process chart, as well as flowchart. For system chart, other terms are run diagram, procedure chart, and flowchart.

The distinction between the flow diagram and the system chart is vital because the use of the standard differs considerably for these two. In the case of the system chart, the focus is upon the inputs and the outputs produced by the sequences of runs, programs, or procedures. In contrast, the focus in the flow diagram is upon the sequences of data transformations needed to produce an output data structure from an input data structure. The flow diagram tells "how." Whereas a system chart identifies programs, runs, or procedures by name and data structures by name, the flow diagram identifies individual operations on portions of data structures. The flow diagram is usually an elaboration of what is indicated by a single process outline in a system chart (see Figure 10).

In the remainder of this paper, system chart conventions and system chart guidelines are considered first. These use a greater variety of outlines, but the logical complexity is relatively low. Then flow diagram conventions and flow diagram guidelines are discussed. Flow diagrams can become logically complex even though the number of different outlines utilized is typically fewer.

System Chart Conventions

Basic format. The basic format of the system chart follows a sandwich rule—that is, it is composed of alternating layers of data identifications and process identifications. The data identifications are equivalent to the bread of the sandwich, and the process identifications are equivalent to the filling in the sandwich. Just as sandwiches may be of the Dagwood type, so the output produced from one process operation may serve as the input for a following process operation (a compound system chart). But a system chart must always begin with inputs (data identifications) and must always end with outputs (data identifications).

To see this sandwich rule in use, con-

[3] The standard's definition of the term "flowchart" is a subversion, well supported by popular usage, of a far older definition. The term "flowchart" has a history predating the use of computers In the field of systems analysis, it historically has designated a graphic aid to analysis quite different from that contemplated in the standard This older use of the term is illustrated in [7, 1963 ed., pp. 237–239], and three forms of flowcharts following this older definition are illustrated in [6, Ch. 5]

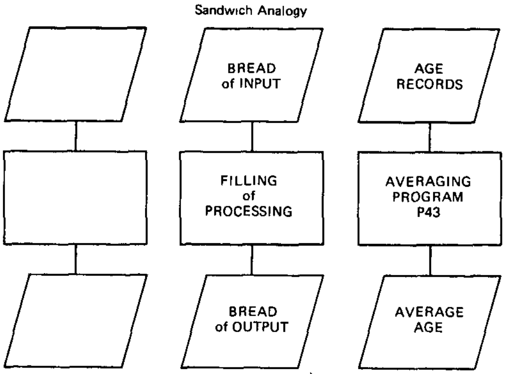

FIG. 11. A simple system chart using only the basic outlines

sider the creation of a system chart using only the basic outlines. Assume that the input available is a set of data about the ages of employees. Assume that the output desired is a single number, the average age of employees. Using the basic outlines, no attention is needed to the media or to the equipment. Hence, as shown in Figure 11, the systems chart begins with an input-output outline for the input. Connected to that by a flowline is a process outline. Connected to that by a flowline and ending the system chart is an input-output outline for the output.

To summarize, the basic format of the system chart is a sandwich. It always begins with the bread of input and ends with the bread of output. The sandwich filling is the processing which converts the input into the output. But the system chart does not tell how the processing is accomplished; it only identifies what processing is done.

Identifications. The bare outlines shown in Figure 11 are meaningful to someone who has the identification of the input, the output, and the processing clearly in mind. But it has less communication value to others because, even though it tells that input is to be converted into output, it does not identify which particular input, what particular processing, or what particular output. To improve the communication value of the system chart, therefore, a common convention is to indicate within the outlines the identification of each input, each output, and each process.

For this purpose, the usual convention is to use the names normally assigned at that installation to the input and output. If the system chart is likely to be read by persons not conversant with those names, then the English-language equivalent in full may be written out within the outlines to provide the identification. Thus, Figure 11 also provides a restatement that incorporates the identifications absent from the system chart on the left in Figure 11.

This same system chart is also present in the top part of Figure 12. But Figure 12 also illustrates more, since it shows a compound system chart rather than just a simple one. The average which was the output of the first processing operation in turn serves as input for another operation. Here the average age of employees is to be combined with previously calculated data on the average age of employees to produce a chart showing the trend of the average age of employees over the course of time. A preparation of this chart is a separate processing operation from a com-

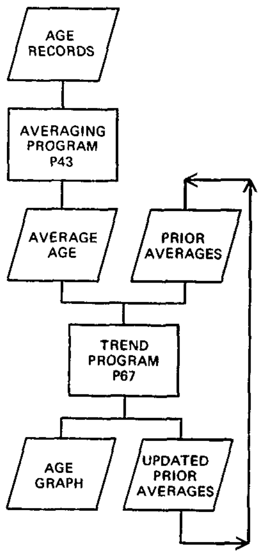

FIG. 12. Compound system chart using only the basic outlines

Case 3:10-cv-03561-WHA Document 693-4 Filed 01/11/12 Page 13 of 29

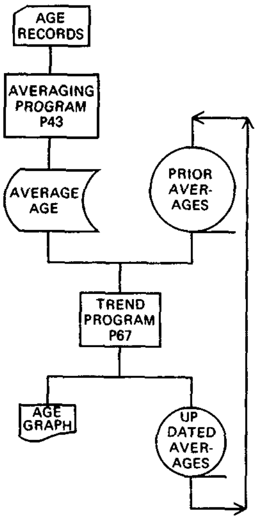

FIG. 13. Compound system chart using the specialized outlines

putation of the current average age. The other output of the trend program is the updated record of the prior averages.

Here, as before, clear identification of each of the inputs, output, and processings is provided, but no indication of the nature of the medium or equipment is provided in the choice of outlines. Note the Dagwood sandwich structure.

Specialized outlines. In order to improve the communication value of the system chart still more, specialized outlines may be used in place of the basic outlines already presented. Thus, Figure 13 presents a redrawing of Figure 12 using the specialized outlines. Figure 13 shows that the data on the age of employees is on cards, and that the output of the average age is put onto a magnetic disk or other external storage device where it serves as input to the trend program.

The other input to this trend program is from a magnetic tape, which has recorded on it the previously computed average ages. A magnetic tape also receives the output from this program so that it can be recycled to serve again as input if necessary.

The hard copy output from the second program is a time series graphic plot. Assuming that the stress is upon the hard copy aspects of this output, then the document outline is the appropriate specialized outline to use. If this plot, however, were displayed on a CRT, then a display outline would be the appropriate choice.

Use of connectors. If a compound system chart requires more space on the page to represent it than is available, then connectors may be used to break the chart into parts and to indicate the connection between the parts on the separate pages. This procedure is consistent with the standard but does not improve the communication value of the chart. An alternative procedure also consistent with the standard gives superior results.

The matter can be likened again to the Dagwood sandwich. If one makes a Dagwood sandwich that is too large to bite, one procedure is to break the sandwich into two or more parts. Whenever doing this, however, one must add an additional slice of bread for each break because each

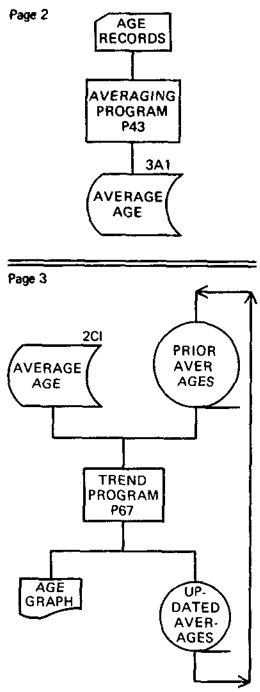

FIG 14 Example of good practice in breaking a system chart

complete sandwich must begin with a slice of bread and end with a slice of bread. Since, by analogy, input or output data serve as slices of bread, one may break a long system chart into shorter charts and still show a connection between them by the choice of representing input or output data. To see this, imagine breaking the system chart in Figure 13 into two separate system charts.

To make the communication value of the system chart high, it is important to show all of the inputs and outputs for each part of the system. If one of these appears at one place as an output, and at another as an input, then it is only necessary to repeat the outline with the identical identification and appropriate cross-reference. This is illustrated in Figure 14. Notice that the disk output from the first run which serves as an input to the second run is shown twice. This is considerably more illuminating to someone who studies any part of the chart than would be the alternative of using a connector outline, illustrated in Figure 15, where two different ways of doing it are shown.

In summary, the most effective way to break a system chart into parts in order to fit it on limited size pieces of paper is to repeat the representation for selected inputs or outputs, identifying and cross-referencing them appropriately. In this way the material shown on each page is complete in itself.

Annotation. When breaking a system chart, a problem arises on identifying the source and use of data. The clearest convention is as shown in Figure 14, by the use of cross-references with an exact repetition of the data identification. This also serves well for multiple uses of an output as inputs to several process outlines.

One alternative way of handling the situation is by the use of the wording within the symbols themselves. But this clutters the space conventionally used for identification, making it a dual-use space rather than a single-use one. This decreases the communication value of the chart, and is not illustrated here.

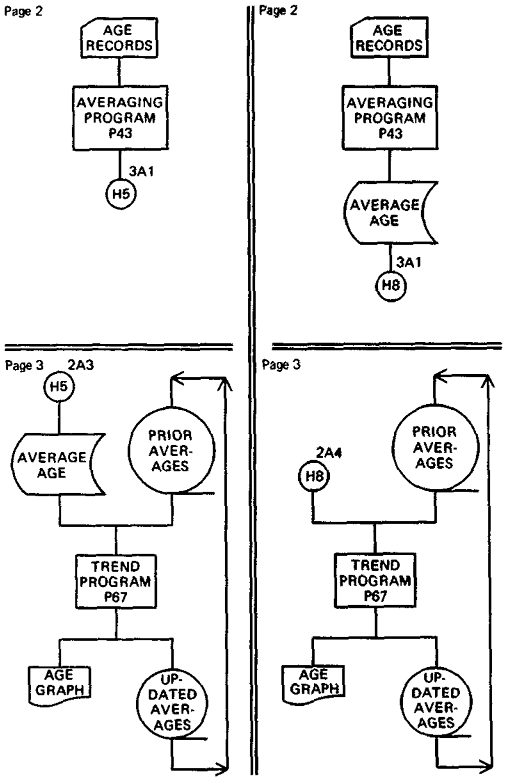

FIG. 15. Examples of poor practice in breaking a system chart

Another alternative way of maintaining the identification of the source and use of data is by the use of an annotation outline. For example, one could be inserted at the bottom of the first page indicating the page to look on to find the additional use of this output as an input, as shown in Figure 16. On the page on which it appears as an input, the annotation outline could again be shown with an indication in it where this input came from as an output. It is clear from even a casual examination of Figure 16 that the excessive use of annotation outlines clutters the system chart and can decrease its communication value.

To avoid the clutter of the annotation outlines yet provide the annotation, an alternative is to add a column of annotation to the right (or the left) of the system chart. This can provide information that is

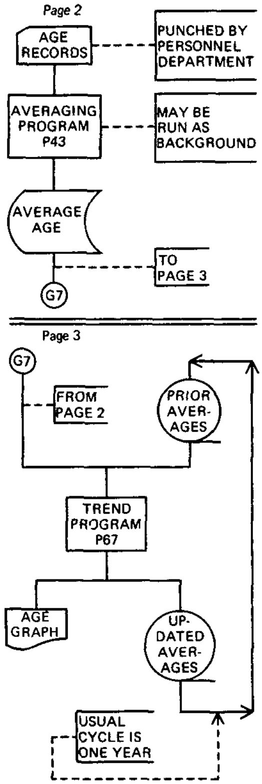

FIG. 16. Annotation symbols in system charts

17, can be widely understood. A chart which uses only specialized names can be fully understood only by those who know the specialized names. Thus, the system chart shown in Figure 19 is less easily understood than the same system chart shown in Figure 17.

A second guideline is to use the data and process identification names consistently and to keep them brief. If the same name appears more than once anywhere in the system chart, it should always identify the same thing.

A third guideline is to use a relatively small size for the outline. It improves the communication value of the chart because it enables a more compact layout, which allows the reader's eye to take in more at one glance.

A fourth guideline is to leave blank space around major unconvergent flows. This visually sets them off, and makes their role in the system more easily comprehended, as in Figure 18. By contrast, in a uniformly packed or tightly spaced sys-

frequently helpful in interpreting system charts; see Figure 17 for an example. The information most commonly shown in such annotation columns is the volume of the input and output, the timing of the availability of input or output, the control procedures, the equipment configurations required, the personnel complements, and the geographic locations. These items are normally not conveniently shown in the system chart itself because they clutter it up too much, and are not of concern to all readers. The use of this column alternative is an elaboration of the standard.

Guidelines for system charts. In preparing system charts, the following guidelines have been found helpful from experience. A first guideline is to choose the wording within the outlines (the data and process identification names) to fit the needs of the readers of the system chart. A chart using something approaching the English language, as illustrated in Figure

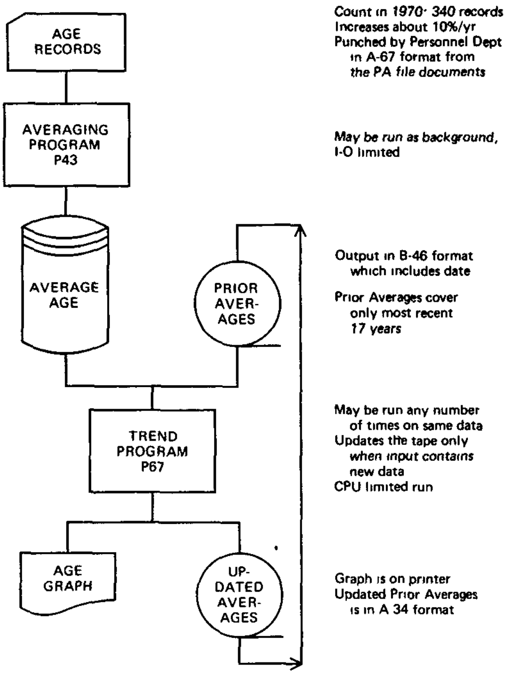

FIG. 17. Annotation column for system charts

Case 3:10-cv-03561-WHA Document 693-4 Filed 01/11/12 Page 16 of 29

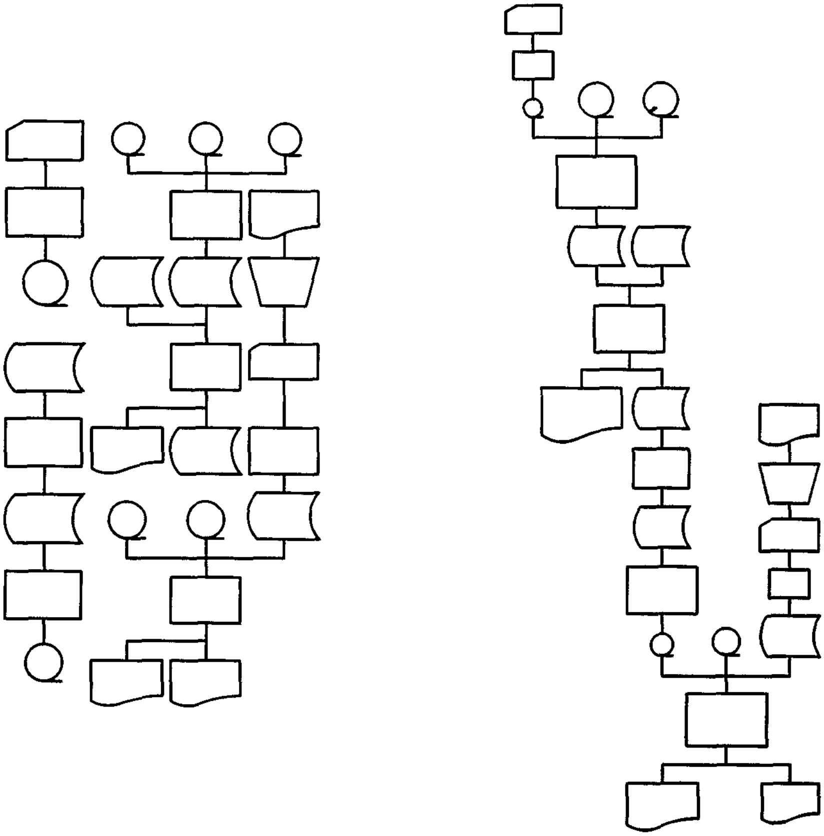

FIG. 18. Example of an open and a tight layout of a system chart for the same system

tem chart, even simple straight lines of flow are difficult to see clearly.

A fifth guideline is to collect incoming flowlines and outgoing flowlines so that the flowlines actually entering and leaving a processing outline are kept to the minimum. This is illustrated in Figure 17.

A sixth guideline is to minimize crossing flowlines. Crossing flowlines can be eliminated by repeating input or output outlines with appropriate cross-references as to source or destination.

A seventh guideline is to use the specialized outlines wherever possible (compare Figures 12 and 13). Their use improves the communication value of the system chart. They are only slightly more difficult to draw.

An eighth guideline is to use cross-referencing and annotation generously, but not to excess. The more the system chart can tell the reader quickly and easily, the more valuable it is. Ways of providing cross-referencing and annotation are mentioned above.

A ninth guideline is to give particular attention to the processing that affects data prior to the time that the data become input to a computer program or run. These are most often manual operations and auxiliary operations. Failure to specify them in full is one of the most common shortcomings of system charts. (Example: How did the data get punched into the cards in the system shown in Figure 19?)

A tenth guideline is to begin with what is known and well understood. Prepare the system chart describing that. Then extend the system chart in each direction. That is, for each of the inputs shown, find the

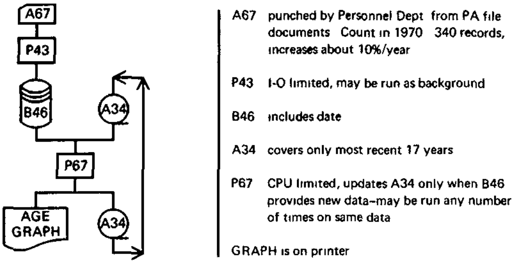

FIG. 19. Example of specialized wording within the outlines of a system chart

processing that produces it and what its input is. Then, regarding those inputs as outputs, continue the procedure for each of those inputs. The same can be done in the other flow direction for the outputs. In this way the system chart can be extended to cover the entire system, as well as covering this system's tie-in with other systems. Typically, particular attention is needed to the manual and auxiliary processing of data in the system.

An eleventh guideline is to make no violations of the standard, and to shun deviations. This requires a clear distinction between a violation, a deviation, and an elaboration. An example may clarify each. Using closed arrowheads on flowlines is an example of a violation. Another example is using a circle with a short right-of-center horizontal line touching the bottom center, as an outline for data on punched cards. This is a violation because the standard assigns a specific significance to such a circle, and provides a specific outline for data on punched cards.

Three examples of deviations from the standard are provided by IBM [14]. IBM advances an outline for "keying." This is a serious deviation since the standard already provides two outlines for data from key-driven equipment, or its operation. These are the manual input (on-line) and the manual operation (off-line). IBM advances an outline for a "transmittal tape." This is a kind of document. The standard provides an outline for documents generally. IBM advances an outline for an "off-page connector." The standard provides a connector outline and specifies the use of cross-references to indicate the location.

An example of an elaboration of the standard is the use of a five-pointed star outline to represent data acquired by the on-line operation of an optical detector of particle tracks in a spark chamber. Such equipment is not generally available, and hence the standard provides no specialized outline. The basic input-output outline is applicable, and hence no specialized outline is really necessary.

Flow Diagram Conventions

Function. The flow diagram describes the algorithm for transforming input data structures into output data structures. As such, the primary emphasis is upon depicting the sequence of operations that tell how data are transformed. The secondary emphasis is upon identifying the portions of the data structures affected and the operations performed. Questions of media or equipment typically become trivial.

Since commonly the operations to transform data structures consist of long sequences of actions, the character of the flow diagram differs considerably from that of the system chart. In the system chart, a sandwich rule describes its basic structure. No such convenient rule serves in the case of the flow diagram. Since it is in effect an elaboration and extension of what is usually shown as a single outline on a system chart, the flow diagram requires many more outlines and a much more extensive presentation of details than does the system chart.

The general character, therefore, of a flow diagram is of a sequence of alternating flowline and process outlines. Somewhere in the early portion of this sequence will usually appear one or more input-output outlines to indicate the input of a data structure. Near the end of the sequence will usually appear one or more input-output outlines to indicate the output of a data structure.

Because of their greater length, the flow diagram must be broken into parts, as a practical matter. For this reason, connector usage and cross-referencing become important considerations in the creating and reading of flow diagrams.

Case 3:10-cv-03561-WHA Document 693-4 Filed 01/11/12 Page 18 of 29

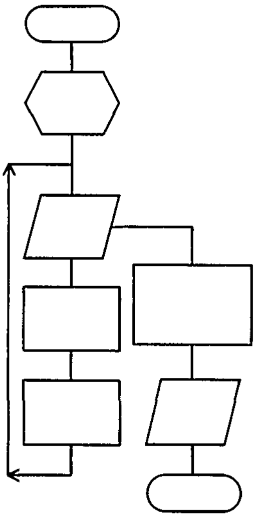

FIG. 20. A flow diagram using basic and additional outlines, with one specialized outline

Basic format. Flow diagrams may be drawn with the basic and additional outlines described previously. For example, consider the program to find the average age of the employees shown in the system chart in Figure 17. Using the basic and the additional outlines, the flow diagram can be stated as shown in Figure 20.

Some comments are in order on this example. First, the beginning and the end of the flow are marked with the termination outline. This is unlike the case of the system chart, where the start or the end of anything was an input-output outline.

Second, the sequence shown follows the common pattern of read-transform-write. Since that transformation cannot usually take place until the input data have been read, the input operation precedes the process operation. Both precede the output operation.

Third, a very common feature of algorithms prepared for implementation on a computer is the use of iteration. This commonly appears in a flow diagram as a loop of flow, as shown in Figure 20. Note that, in order to indicate a section of flow contrary to the normal rule, a long flowline together with open arrowheads has been used.

Just as in the case of the system chart, a flow diagram using only the basic outlines provides some information but not as much as it can when augmented with written identifications within the outlines. These identifications are important to indicate the portions of the data structures affected and the operations. A long sequence of process outlines which do not designate the portions of the data structures affected soon becomes ambiguous. For this reason it becomes, as a practical matter, important to specify in detail in a flow diagram exactly what portions of the data structures are affected and in what way.

To this end, the flow diagram in Figure 20 can be redrawn as shown in Figure 21. Here the outline choice and sequence are identical to those in Figure 20, but now the outlines each contain identifying information. The identifications consists of four things: the names of parts of the algorithm as implemented (entry points, usually, such as THRD), the names of operations (such as "add"), the names of conditions (such as "end of data"), and the names of operands (parts of the data structures affected, such as TOTAL).

Specialized outline. To improve the communication value of a flow diagram, the usual practice is to use the specialized process outlines wherever possible. The

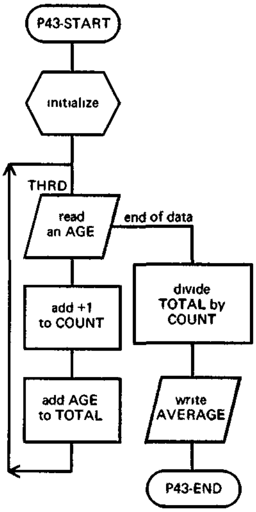

FIG. 21 A flow diagram with identifications

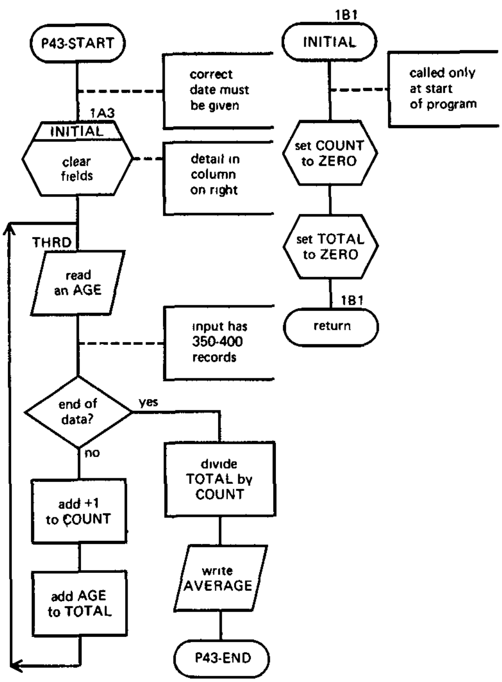

FIG. 22. A flow diagram using specialized outlines

specialized process outlines available for this purpose are the decision, preparation, predefined process, merge, extract, sort, and collate outlines. The manual and auxiliary operation outlines are not usually applicable because the flow diagram commonly represents only processing operations carried on within a computer.

Using these outlines, Figure 21 can be redrawn as shown in Figure 22. Here the decision outline provides an explicit end-of-input-data test. The striped preparation outline refers to the short flow sequence called INITIAL.

Also available in the standard, as described previously, are specialized input-output outlines such as those to indicate media or equipment. As mentioned previously, the main attention in a flow diagram is not upon the media or the equipment but upon the logical character of the data transformations. Use of the specialized input-output outlines clutters the flow diagram with additional shapes which tend to distract from the main focus of attention. Further, with present-day operating systems, the equipment and media for input and output can be altered at any time for operator convenience, and hence are neither statically nor dynamically determined by the character of the data. For these reasons, the use of specialized input-output outlines in a flow diagram is not recommended. Their use is not a violation of the standard, but contributes little to the communication value of a flow diagram and can even detract. For this reason, their use is not illustrated in this paper.

One outline that is illustrated is the annotation outline. This outline can be very helpful in the flow diagram to describe values and to provide explanation. Thus, in Figure 22, an annotation outline has been used to indicate, for the iterative loop, the expected number of times the loop will be executed. This information, it should be noted from Figure 21, is not available from the outlines or from the normal identification information supplied within the outline. Another use of annotation outlines is the warning about the need for the accurate date of the run, as shown in Figure 22.

Connectors and cross-references. Connectors and cross-references are important in flow diagrams because of the common length of flow diagrams and because of convergent and divergent flows. Flow diagrams are almost always too large to represent on one sheet of paper. Usually they include alternative flow paths. Convergence points (fan in) and divergence points (fan out) must be presented and the flows clearly identified.

To make the communication value of a flow diagram high, it is desirable to have the flow pattern shown in as linear or straight-line a form as possible. The more cut up, chunky, bunched, or branched the flow pattern is, typically the more difficult it is for a person to comprehend. For this reason, the use of connector outlines and of cross-referencing normally helps give a smoother, more linear appearance to the flow.

If the flow diagram shown in Figure 22 could not be represented all on one page but had to be broken into parts, it could

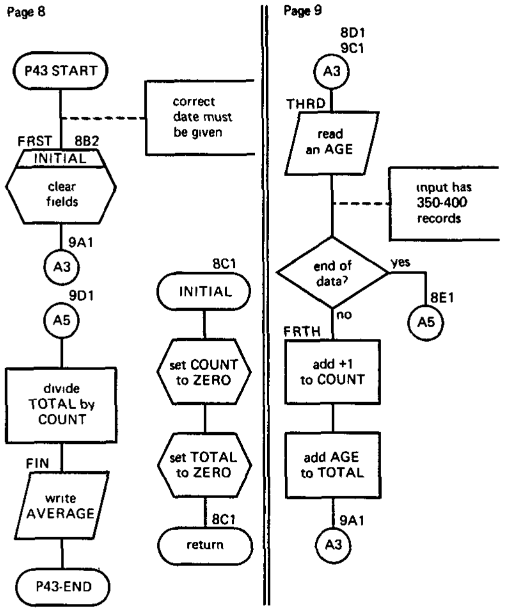

FIG. 23. A flow diagram with connectors and cross-references

be broken at any point. One way of doing it is shown in Figure 23. The connector outline does not substitute for any other outline but instead serves as an additional outline, in effect specializing the flowline symbol.

As one reads the line of flow down the page 8 part of Figure 23, one encounters an exit or outconnector A3. The reader can then search Figure 23 for an entry or in-connector having the same identifying set of characters A3 within it. This entry or inconnector is the continuation of the flow. The identification within the entry and exit connectors must permit unequivocal and unique identification of the one appropriate entry connector to be associated with each exit connector.

To facilitate finding the entry connector for each exit connector, the common practice is to use cross-references in the manner described previously If any name has been assigned to a portion of the algorithm or program (that is, the program data structure), such as THRD, then this name may be entered above and immediately to the left of the connector, or within the connector itself If a coordinate plan has been established for identifying portions of the flow diagram, then the location indication can be entered above and immediately to the right of the connector, as shown in Figure 23.

The standard does not specify any particular coordinate convention for use specifically in flow diagrams. As a practical matter the most common convention used is a page number followed by a row and then a column designation (see Figure 24 for an example). Some omit the row designation and some omit the column designation Some use a sequential count to provide an index of the position within a row or within a column of a connector position. Anything that begins with a digit causes no confusion since names in most programming languages may not start with a numeral.

The symbolic name and cross-reference notations may be used with any outline, not just with connectors. They are needed for all horizontally striped outlines, as a minimum, as shown in Figure 23. The names also provide a way of cross-referencing any part of the flow sequence shown in the flow diagram to the actual program coding. This is illustrated in Figure 23 by the names FRTH and FIN, for example, which refer to portions of the program

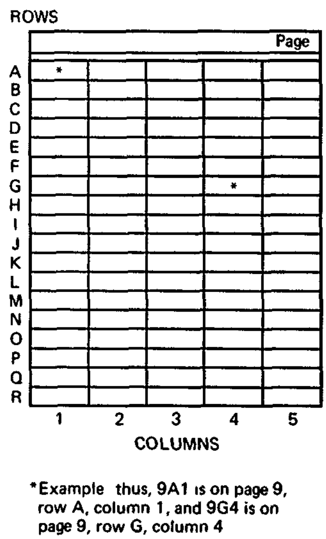

FIG 24 One cross-referencing scheme

:= ←	is replaced by	≠ ¬=	is not equal to
+	plus, or addition	≮ ¬<	is not less than
−	minus, or subtraction	_	underline, blank
* ×	multiplication	$\vert\ \vert$	absolute value
** ↑	exponentiation	¬ ∼ −	negation
/	division	$\vert$	logical OR
EOF	end of file	&	logical AND
:	comparison	' '	literal (zero level address)
>	is greater than	( )	grouping, level of address
=	is equal to	A( )	address constant
<	is less than		
≯ ¬>	is not greater than		

FIG. 25. Symbols for use in flow diagrams

respectively, even though they are not entry points.

Notation. The standard does not specify any particular language, symbols, or notational scheme for use within the outlines to identify the data or to name operations. Ordinary Enlgish prose tends to be too verbose to be readily accommodated within small-size outlines. Yet using small-size outlines yields a more easily comprehended flow diagram. A notational scheme or set of symbols that is compact and that permits easy representation of the common situations is highly desirable.

To this end, some years ago the American Standards Association (now ANSI) circulated a working paper advocating a notation or set of characters (graphics) for such a use (references). Many users of flow diagrams prefer to use the same notation they use in the programming language. Although this works well with some computationally oriented languages on computationally oriented jobs (such as FORTRAN or PL/I in engineering or scientific work), it falls short of the need for string operations, complex operations on arrays, and manipulations of all types of structures in logical terms. To meet these problems, several notational schemes have been advanced in the literature, of which the Iverson notation is probably the most widely known [16].

An eclectic list drawn from these three major sources is offered in Figure 25. This list is composed from graphics included in the ASCII, EBCDIC, and IBM BCD codes.[4] Hence, computers can print these in computer drawn flow diagrams. The one exception is the arrow, since none of these codes includes arrows. The arrow, however, is still widely found where communications and display equipments are used, and has a history of use in programming work. It provides a neater alternative, especially for publication, than does a double graphic.

Of particular importance are the "is replaced by" symbols. Two alternative symbols are common, and serve to indicate that the symbol on the left has its value determined by what is on the right, as illustrated in Figure 26. For this purpose, the equal sign has sometimes been used. Such usage is inconsistent with mathematical practice, and gives a dual role to the equal sign, the other role being to indicate equality, as for example, from a comparison.

For comparison, a colon is common. The variable of comparison is shown on the left, and the standard or constant of comparison is shown on the right side of the

[4] American Standard Code for Information Interchange, published by ANSI, Extended Binary Coded Decimal Interchange Code; and Binary Coded Decimal.

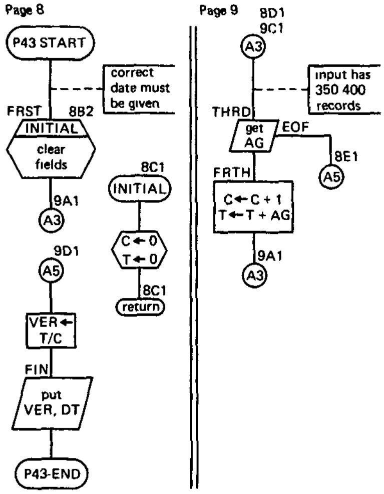

FIG. 26. The flow diagram of Figure 23 redrawn using symbols within the outlines

colon. The exit flowlines from a decision outline must be provided with an indication of the basis for their choice, expressed in terms consistent with the notation used within the decision symbol outline, as shown, for example, in Figure 27.

Parentheses can indicate grouping, a usage borrowed from mathematics. Another use is to indicate levels of addressing. Most literals are enclosed within prime marks (single quote marks) to indicate the zero-level addressing status. Numeric literals, when they are to be used in arithmetic operations, are sometimes shown without enclosing prime marks. Nonnumeric character combinations appearing without the prime marks are assumed to be first-level addresses—that is, the names of items of data, such as a field or variables.

The notation for second and higher levels of addressing is to enclose in successive pairs of parentheses, one for each additional level of addressing desired. A special variant of this is the address constant—that is, something whose value will be determined by its machine language address at the time of execution. An A in front of the parenthesis can serve this purpose.

The use of a terse notation such as summarized here permits considerably greater amounts of material to be shown within each outline, or smaller size outlines to be used in a flow diagram. In both cases, an improvement in the communication value of the flow diagram typically results.

Guidelines for flow diagrams. A first guideline is to chose the wording or symbols within the outlines to fit the readers of the flow diagram. This depends in a major part upon the level of detail to be shown. The more summary (less detailed) this is, the more difficult is it to find a satisfactory wording or symbols to use within the outlines. As a general rule, whatever is chosen should be terse in order to permit the use of small size outlines.

A second guideline is to be consistent in the level of detail shown in the flow diagram. If some parts of the flow diagram are in great detail and others are only sketchy, the statement of the algorithm is distorted. A consistent level of detail provides a sounder basis for making judgments about the algorithm and presents a better basis for making estimates of computer time, programming time, and conversion difficulties and for debugging than does a fluctuating level. Maintaining a consistent level of detail is simple only when the level of detail matches the implementing programming language. The difficulty comes with flow diagrams at summary and intermediate levels of detail.

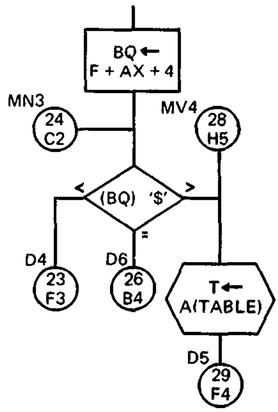

FIG. 27. Entry and exit flowlines in a flow diagram

A third guideline is to use identifying names consistently. Given the type and level of symbols for use within the outline, names for data and operations should be used uniformly and accurately. Figures 21, 26, and 27 illustrate good practice, for example.

A fourth guideline is to use cross-references liberally in the flow diagram. Cross-references to both the program and to locations in the flow diagram improve the communication value of the flow diagram. To keep connectors small and to use space efficiently, it is helpful to use a location cross-reference inside the connector outline, and the program name cross-reference outside the connector, as illustrated in Figures 26 and 27.

A fifth guideline is to keep the flow diagram simple and clean. Clutter and lack of "white space" decreases the communication value of the flow diagram. For convenience and clarity, spacing the diagram so it can be typed (if it is prepared by manual means) is a real assistance.

A sixth guideline is to keep clearly separate the operations to be performed on program data structures from those to be performed on operand data structures. That is, operations on the program itself, such as switch settings, indexing, initialization of program control variables, and the like, should be shown in preparation outlines, separate from operations that transform input into output data. This guideline helps make a flow diagram more easily understood, and it improves debugging work. To facilitate a clear separation, some people add a mark to the basic outlines that specify operations on the program data structure. If used, the mark should be one that is easily detected in a quick scan of the flow diagram. Examples are a large round spot within the outline, or a bold left edge for the outline, or a shaded upper left or center corner of the outline. None of these techniques are sanctioned by the standard; only the second is a violation of the standard, and hence should not be used.

A seventh guideline is to avoid using successive connector outlines. If more than one connector should properly appear in a series (as when multiple names are assigned to one entry point, or when the program calls for consecutive unconditional transfers of control), good practice is to collect the connectors to the left or above the line of flow. A tree arrangement (like an upside-down version of that used for multiple exits from a decision outline, as shown in the top center of Figure 6) can also be used.

An eighth guideline is to observe consistently the general flow pattern from top to bottom and from left to right. When this guideline is observed, arrowheads may be omitted from the flowlines that conform to the guideline. Clarity in a flow diagram is improved by arranging the main flow to conform as much as possible to the guideline. Minor and alternative flows may then deviate from normal, and by this deviation can be identified as not part of the main flow.

A ninth guideline is to draw entrances at upper left and exits at lower right. Entry and exit connectors are most noticeable if they are in a consistent position Thus, the usual practice is to place an entry either above or to the left of the line of flow it is to join, and to place an exit either below or to the right of the line of flow it comes from, as shown in Figure 27. For both cases, when set to the left or right, then the flowline goes from left to right. If set above or below, the flowline drops down to or from the line of flow. Partial exceptions may be made to maintain symmetry, as when a decision outline has three exits, as exampled by exit D4 in Figure 27.

A tenth guideline is to draw flowlines so that they enter and exit at the visual centers of the outlines. The outlines usually possess either vertical or horizontal symmetry, typically about their center points. If therefore entrance and exit flowlines be drawn vertically and horizontally so that they appear to point toward or emerge from the center point of the sym

bol, the visual appearance of the flow diagram is improved.

An eleventh guideline is to use connectors and cross-references to avoid excessive crossing flowlines. The reasons for this have already been presented above.

A twelfth guideline is to draw only one entrance flowline per outline symbol. This is especially important in summary and intermediate levels of flow diagram. When more than one operation is specified or implied within an outline symbol, multiple entrance flowlines could mean that the sequence of events within the outline symbol may be different for the respective entrances, or even that not all the operations are to be performed for one or more of the entrances. To avoid this ambiguity, the usual practice is to make the flowlines join prior to entry into the outline, as shown in Figure 27. An alternative approach lacking conformity to the standard and not free of ambiguity is to subdivide an outline (see, for example, [9]). When the notation within the outline symbols is at the same level of detail as the program itself, no practical difficulty arises. For consistency, however, this guideline should be observed at all levels of detail.

A thirteenth guideline is to draw, with four exceptions, only one exit flowline for each outline. One exception is the decision outline which by definition must have multiple exits. An example is shown in Figure 27. The second is the input-output outline when an end-of-file or end-of-data condition may result in a failure to read input data. The notation within the outline should specify clearly the basis for the choice of the exit. The other exceptions are the extract and collate outlines in system charts.

A fourteenth guideline is to identify clearly all multiple exits. This is an extension of the previous guideline, and is illustrated in Figure 27.

A fifteenth guideline is to make no violations of the standard, and to shun deviations. This guideline is explained above.

Preparation of flow diagrams. The preparation of flow diagrams is primarily a heuristic matter. It is one for which no hard and fast rules can be presented.

A basic heuristic approach to the preparation of a flow diagram uses the "in-do-out" pattern. In order for the computer to produce output data, it must have some input data to operate on. It cannot perform the processing until the input data has been made available. Hence, the general character of a flow diagram is a sequence involving the following steps: preparing to read input data, reading input data, processing the input data read, producing the output data, and ending the processing actions.

This is sometimes referred to as initialize, process, and cleanup. But in turn, each stage itself has these elements usually also present. Thus, in order to do the output operation, it is necessary to do some initialization, to do the actual output, and then to do a cleanup.

A second common heuristic approach to assist in preparing flow diagrams is to assume that a job is really much simpler than it actually is. Then a flow diagram is prepared for this simpler job. After that, more realism is added by dropping some of the simplifying assumptions, and the flow diagram is redrawn to reflect this closer approximation to the real job. This two-step process is repeated as many times as needed. Each successive flow diagram comes closer to the actual job.

A third common heuristic approach is to draw a series of flow diagrams, each one at a greater level of detail. The first flow diagram is very general and provides only an overview (a "first cut" at a job). Then each successive flow diagram amplifies and provides additional detail. Each expands each of the outline symbols in the earlier flow diagram into a sequence of outline symbols. The final diagram, then, is near or at the same level of detail as the programming language that is to be used.

PROBLEMS IN THE USE OF THE STANDARD

The evidence is clear that thus far, people in the computer field do not use the standard. For example, the author of this paper made a survey of the 1968 and 1969 issues of five periodicals, sixteen books published in 1968 and 1969, a sample of a dozen computer installations, and a sample of five flowcharting software packages. The evidence from each supported the others.

The periodicals surveyed had surprisingly few articles using flowcharts. The periodicals were the *Communications of the ACM*, *Computers and Automation*, *Datamation*, *Business Automation*, and the *Journal of Data Management*. The books included texts on programming, and conference proceedings, as well as specialized and general survey books. The installations covered a range of sizes from small to very large. The software packages included the best-known flowcharting programs.

The conclusion was that none of the installations even attempted to use the standard, and none of the periodicals had any article free of violations of or deviations from the standard. Only one of the books used the standard free of significant deviations, but even it used the 1963 version of the standard. Only one of the flowcharting software packages offered the standard, and that only as a nondefault option Most flowcharts observed in the survey followed the computer vendor recommendations, especially the 1964 recommendations from IBM.

This lack of use of the standard suggests the existence of a number of problems. These fall in four main areas: ignorance, construction, conception, and applicability. A brief look at each may help clarify the reasons behind some of the material presented earlier in this paper.

Ignorance of the standard is widespread. This ignorance has generally been abetted by the professional societies, the computer vendors, and ANSI. For example, ACM, which even has a policy of supporting standardization work, has published for the information of its membership and for the guidance of the professional community, only the 1963 version of the standard. It has published to date none of the revisions. Nor does it require that all flowcharts in its publications conform to the standard. The action of the computer vendors is noted above. It is noteworthy that IBM in 1969 began the issue of a new template [14] that can be easily used to prepare flowcharts conforming to the standard. ANSI, by its refusal to grant free reprint rights and by its high charges per page for copies of the standard, severely inhibits the spread of knowledge about the standard. This is a case where ANSI's need for income to support its work results in hindering its work.

Some of the construction problems are closely related to the ignorance problem just noted. The two most common violations of the standard are in the ratio dimensions of the outlines and in the use of deviant outline shapes. Some would-be users of the standard are just ignorant; others in an attempt to make better use of space are willful violators. In the latter respect, the two most common complaints are that the standard wastes space and that it fails to provide enough space. People cite the "seven-page program that takes a twenty-page flow diagram" and claim it is faster and easier to read the program listing. Although cross-referencing in the diagram helps, it is not the entire solution. Summarization is a solution, but this gets into conception problems. On the charge of wasting space, many would-be users of the standard fail to draw outlines of differing size (while preserving the ratio dimensions) to fit the wording or symbols that go within the outline. This is especially true for decision outlines and connectors, as a practical matter.

The conception problems are more serious. Thus, it is widely agreed that a flow diagram is most useful if it is not as detailed as (is more summary than) the program it describes. To create these

requires compressing, condensing, and eliminating details. But which ones? And how many? A poor choice can render the resulting flow diagram nearly useless. Being more summary also increases the difficulty of providing useful cross-referencing between parts of the flow diagram and between the diagram and the program.

This level of detail problem affects the rigor and completeness of a flowchart. Prepared in full detail, everything must be present in its proper place. All ties together. When full detail is absent from the flowchart, it becomes difficult or even impossible from the flowchart itself to determine whether a particular process or operation is correctly shown or is essential. This difficulty arises particularly for decision outlines in flow diagrams and for process outlines in system charts.

The applicability problems are major. A minority of people in the field hold that a flowchart is a waste of time. Since neither programmers nor analysts, they claim, think in terms of flowcharts, why prepare them? Preparing and trying to read them just divert people from a more productive use of their time. It is this author's observation that this indeed is true for some people, but the documentation value of flowcharts for other people's use still justifies their preparation and maintenance. Along this same line, others claim that alternative techniques offer superior documentation, for example, decision tables [8], logic flow tables [20], Ivenson notation [16], or networks or directed graphs. Further, conceptually the standard could be applied to prepare other types of graphic aids than the system chart and the flow diagram. So far, these have yet to be defined in the literature in clear form.

These objections and alternatives have indeed a measure of truth. This can be seen in several aspects of the flowchart. First, the flowchart is weak in showing the timing of processing and of data availability or need. This lack of showing "when" is very serious for communications oriented systems and programs, for example. Second, the flowchart does not directly tell "why"; it tells instead "what" and "how." It leaves the "why" to human inference. Third, the flowchart does not tell "how much." This cannot usually even be inferred. Fourth, the flowchart does not fully tell what or who does or is to do something. It has elaborate outlines for identifying peripheral equipment and media, but no way of indicating people, other computers, or parallel or alternative parts of a given computer as performers of actions on data. It is doubtful if any attempt should be made, beyond the use of annotation, to extend the flowchart beyond what it is already. It cannot be all things to all analysts and programmers; but it does have a place.

CONCLUSION

Both the system chart and the flow diagram provide important tools for the analyst and the programmer in preparing and in documenting work for implementation on a computer. Carefully prepared, flowcharts can enhance the rigor with which the analyst and programmer think through the systems, programs, and associated procedures. This greater rigor in turn typically reduces the cost of debugging. But the major contribution of the system chart and the flow diagram are to communicate some essential aspects of information processing work from one human being to another. The quality of this communication is enhanced by a consistent use of the standard.

APPENDIX. DRAWING OUTLINES

Terms

Ratios. The dimensional ratio of the outlines defined in the standard is determined by the following procedure. Construct a rectangle circumscribing the outline. The rectangle must be formed from vertical and horizontal lines, and each line should just touch the inscribed outline. The ratio is then determined by measuring the horizontal and the vertical sides of

the circumscribing rectangle, and is usually defined as the height as a proportion of the width taken as unity.

Sizes. The standard permits outlines of any size as long as the dimensional ratio and general configuration are maintained.

Lines. The standard indicates the use of any uniform width or weight of line for all of the outlines, irrespective of their sizes. Lines need not be continuous, but may be created by the close spacing of discrete symbols.

Configuration. The standard specifies that the outlines in shape should conform sufficiently close to the configurations shown in the standard to permit rapid and unambiguous identification. The curvature of lines and the magnitude of angles may vary from those shown in the standard, provided that the shapes still be clearly recognizable. To that end, since the angles and curvatures shown in the standard are sometimes difficult to use in drawing, the configurations and description given in this paper sometimes simplify or round off to make the drawing easier, within the restrictions imposed by the standard.

Orientation. The figures in this paper illustrate the general orientation specified by the standard for each outline. Outlines may not be turned. Each outline except the connector and the decision outlines has at least one straight line which must be either vertical or horizontal, as illustrated in the figures. The connector outline has no specified orientation; the decision outline is horizontally oriented along its widest dimension. Flowlines should normally be vertical or horizontal, and may make right-angle bends. Flowlines deviating from horizontal or vertical are neither recommended nor proscribed by the standard.

Outlines

Input-output. This is a parallelogram with its base edge 15 degrees further to the left than its top edge. It has a width-to-height ratio of 1 to $\frac{2}{3}$, as illustrated in Figure 1.

Process. This is a single rectangle with a width-to-height ratio of 1 to $\frac{2}{3}$, as shown in Figure 1.

Flowline. This may be of any length. The spread between the barbs of the arrowhead and the length of the arrowhead are each approximately ten times the width of the flowline, as shown in Figure 1. The angle of the barbs is $26\frac{1}{2}$ degrees from the flowline—i.e. the entire arrowhead just fits within a square.

Annotation. This is a rectangle with the right side missing, but with a width-to-height ratio of 1 to $\frac{2}{3}$, as shown in Figure 1.

Connector. This is a circle, as shown in Figure 2. In practice, it should be kept as small as possible. A size just large enough to accommodate four characters of wording is about typical.

Terminal. This has a width-to-height ratio of 1 to $\frac{3}{8}$ and looks like a slim rectangle but with half-circle ends, as shown in Figure 2.

Parallel mode. This is two horizontal parallel lines of any equal length spaced about ten or so line widths apart, as shown in Figure 2. Entering the upper one may be one or more vertical flowlines positioned anywhere along the line. Leaving the lower one may be one or more vertical flowlines positioned anywhere along the line. The number of entrance and exit flowlines may not both be equal to one and neither may be equal to zero—i.e. one must have two or more flowlines, and one must have one or more flowlines.

Document. The top three edges are a portion of a rectangle, but the bottom edge is a curved line to represent a break in the paper. The width-to-maximum-height ratio is 1 to $\frac{2}{3}$, as shown in Figure 3. The width-to-height ratio of the rectangle portion is 1 to $\frac{1}{2}$. The radius of the left curve is one-half the width, and the center is one-fourth of the width in from the left edge. The radius of the right curve is one and one-quarter of the width, and the center is straight below the right edge of the outline.

Magnetic tape. This is a circle with a horizontal rightward pointing line tangent to the bottom. The tail line, as shown in

Figure 3, extends over to the point of intersection with an imaginary vertical line tangent to the rightmost point on the edge of the circle.

Punched-card. The punched-card symbol has a width-to-height ratio of 1 to $\frac{1}{2}$ and appears generally like an upper left corner cut punched card, as shown in Figure 3. The corner cut has an angle of about 30 degrees and cuts off about one-sixth of both the width and height.

Punched-tape. This has a maximum-width-to-maximum-height ratio of 1 to $\frac{1}{2}$, as shown in Figure 3. The centers of the curves are at the one-fourth points of the width; the radius of the curves are three-quarters of the width. The end lines are vertical. The top left corner and the bottom right corner are not more than 10 percent of the height in from imaginary horizontal lines tangent to the maximum points of the arcs.

Display. As shown in Figure 4, the width-to-height ratio is 1 to $\frac{2}{3}$. The radius of the curved lines is one-half of the width. The neck curves join the horizontal lines one-third of the width in from the left ends.

Manual input. This outline, as shown in Figure 4, is a stylization of a cross section of a keyboard with the sloping surface (at an angle of about 10 degrees) of the keyboard having its lowest point to the left. This cuts about one-third off the height of the left vertical edge. The width-to-maximum-height ratio is 1 to $\frac{1}{2}$.

Communication link. As shown in Figure 4, the zigzag has an angle of 45 degrees. The distance between the parallel line segments is from ten to fifteen times the width of the flowline. The lines may be of any length, and one or more zigzags may be located anywhere along the lines.

On-line storage This is a stylization of a portion of a cylinder with a convex end at the left and a concave end at the right, and having a width-to-height ratio of 1 to $\frac{2}{3}$ as shown in Figure 4. The ends are arcs with a radius equal to one-half of the width.

Disk storage. This outline shown in Figure 4 is a reorientation of the on-line storage outline, but with three lines added, one convex on the top, and two to mark off bands. These are arcs with a diameter equal to the width of the equivalent on-line storage outline. The spacing of the bands is about one-tenth to one-twelfth of the width of the equivalent on-line storage outline. The overall width-to-height ratio is 1 to $\frac{5}{8}$.

Drum storage. This outline, as shown in Figure 4, omits the two band arcs from the disk storage outline, and reorients it to match the on-line storage outline. The overall width-to-height ratio is $\frac{5}{8}$ to 1.

Core storage. As shown in Figure 4, this is a square and hence has a width-to-height ratio of 1 to 1. The two lines, each parallel to a side, are in one-eighth of the width from the edge of the outline.

Off-line storage. This outline is an equilateral triangle standing on a point. Since it is equilateral, it has a width-to-height ratio of 1 to 0.866 (about 15 to 13), as shown in Figure 4. The small line drawn about six-tenths or eight-thirteenths of the distance from the top to the bottom tip of the triangle is a required part of the outline.

Decision. As shown in Figure 5, this is a diamond outline with a width-to-height ratio of 1 to $\frac{2}{3}$.

Preparation. The horizontal lines, as shown in Figure 5, are two-thirds of the total width. This makes the angle of the four sloping sides fall at $26\frac{1}{2}$ degrees from the vertical, since one-sixth of the total width is missing from each end of the horizontal lines. This gives a width-to-height ratio of 1 to $\frac{2}{3}$.

Predefined process. This is a rectangle with vertical bars in about one-eighth to one-sixth of the width from the left and right ends. Figure 5 shows the lines at the one-eighth position. The overall shape is the same as the basic process outline with a width-to-height ratio of 1 to $\frac{2}{3}$.

Manual. This is a keystone (trapezoidal) shaped outline, as shown in Figure 5, with a maximum width-to-height ratio of 1 to $\frac{2}{3}$. The slope of the sides is

about 15 degrees. This makes the length of the bottom horizontal line equal to one-half of the width.

Auxiliary. This outline as shown in Figure 5 is a square.

Merge, extract, sort, and collate. These are shown in Figure 5. All are constructed from equilateral triangles. Those that use two abutting triangles must use triangles of the same size for the two parts.

Card deck. As shown in Figure 9, this is an extension of the punched-card outline. The left corner outline is extended to give a height about one-fifth greater than that of the punched-card outline. The topmost horizontal line is extended about one-eighth of the width beyond the right end of the punched-card outline embedded in this outline, and the new right vertical is made slightly shorter, as shown in Figure 9. This gives an overall width-to-height ratio of about $\frac{5}{4}$ to $\frac{2}{3}$.

Card file. This is like the card deck outline but with parts of the top, left slant (corner cut), right slant, and right vertical erased. The position of the top and right lines is determined in the same manner as for the card deck outline. The remaining upper right right angle has lines equal in length to about one-half of the height of the punched-card outline embedded in this outline.

REFERENCES

1. AMERICAN STANDARDS INSTITUTE.[5] Proposed American Standard flowchart symbols for information processing *Comm ACM 6*, 10 (Oct. 1963), 601–604
2. ANSI Standard flowchart symbols and their use in information processing (X3.5) American National Standards Institute, New York, 1970 (20 pp.).

[5] As of 1969, American National Standards Institute (ANSI).

3. —— Standard graphic symbols for logic diagrams (Y32.14). American National Standards Institute, New York, 1962 (12 pp.).
4. ——. Standard method of charting paperwork procedures. American National Standards Institute, 1959 (12 pp.).
5. USA STANDARDS INSTITUTE.[5] Graphic symbols for problem definition and analysis—a standards working paper. *Comm ACM 8*, 6 (June 1965), 363–365.
6. ASSOCIATION FOR SYSTEMS MANAGEMENT. Charting. In *Business Systems*, Assoc for Systems Management, Cleveland, Ohio, 1963, Ch. 5.
7. CHAPIN, NED. *An Introduction to Automatic Computers* Van Nostrand, Princeton, N J, 1957 (525 pp.) and 1963 (503 pp.).
8 ——. An introduction to decision tables. *Data Proc Manag. Ass. Quart. 3*, 3 (April 1967), 2–33.
9 FORSYTHE, ALEXANDRA I., ET AL *Computer Science*. Wiley, New York, 1969 (553 pp.).
10 FRITZ, W. BARKLEY, ET AL. (SHARE's AD HOC COMMITTEE ON FLOW CHART SYMBOLS). Proposed standard flow chart symbols. *Comm ACM 2*, 10 (Oct. 1959), 17–18.
11. GOLDSTINE, H. H, AND VON NEUMANN, JOHN. *Planning and Coding Problems for an Electronic Computing Instrument, Vols I, II, III.* Van Nostrand, Princeton, N.J., 1947, and 1948.
12 GRUENBERGER, FRED J, ET AL. *Introduction to Electronic Computers.* Wiley, New York, 1963 (167 pp.).
13. IBM CORPORATION. Flowcharting techniques, C20-8152. IBM Corp, New York, 1964 (34 pp).
14 ——. Flowcharting template, X20-8020. IBM Corp., New York, 1969 (one plastic cutout drawing guide in a printed envelope).
15. —— Problem planning aids, IBM Type 650. IBM Corp, New York, 1956 (18 pp.).
16 IVERSON, KENNETH E. *A Programming Language.* Wiley, New York, 1962 (286 pp.).
17. LEWIS, F. DAVID, ET AL. Program 1401-2.0.019 IBM Corp. New York, 1963 (18 pp.).
18 MCCRACKEN, DANIEL D., ET AL *Programming Business Computers.* Wiley, New York, 1959, pp. 26–43 and *passim.*
19 RCA SERVICE COMPANY. The Language and Symbology of Digital Computer Systems RCA, Camden, N J, 1959 (114 pp.)
20 SELF, SIDNEY B Logic flow table. *J. Data Manag 5*, 12 (Dec 1967), 30–36
21. UNITED STATES AIR FORCE AFICCS Documentation Standard Manual TEM-AF-2. US Air Force, Washington, D.C., 1968 (126 pp)